



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

mk

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/662,034

09/12/2003

Paul J. Mantey

200309970-1

9298

22879

7590

12/05/2006

EXAMINER

SPITTLE, MATTHEW D

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

ART UNIT

PAPER NUMBER

2111

DATE MAILED: 12/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/662,034	Applicant(s) MANTEY ET AL.	
	Examiner Matthew D. Spittle	Art Unit 2111	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 October 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-13 and 42-44 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-13 and 42-44 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claims 1 – 13, and 42 – 44 have been examined.

Claim Rejections - 35 USC § 102

5 The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

10 (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1 – 4, 12 and 43 are rejected under 35 U.S.C. 102(b) as being anticipated by Johnson et al. (U.S. 6,122,758).

Regarding claim 1, Johnson et al. describe a computer system comprising:

15 A system bus implemented in accordance with an Inter-IC bus specification (Figure 4, 7, item 310; column 7, lines 10 – 12);

A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65);

20 A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32);

A first first-in first-out (FIFO) buffer coupled to the send machine, the first FIFO further coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 – 10)

25 over an internal bus (Figure 7, item 226) but not over the system bus, the first FIFO not

Art Unit: 2111

being coupled to the send machine over the internal bus (Examiner notes that FIFO (516) is connected to the send machine (707) over intermediate DATA bus and not over the internal bus (226).

30 Regarding claim 2, Johnson et al. describe the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but
35 note that the interrupt is not generated until a response is received from the I2C device).

Regarding claim 3, Johnson et al. describe wherein:

The first FIFO buffer comprises means for receiving a plurality of bytes from the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both
40 queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes);

The send machine comprises means for transmitting the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and
45 processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

Art Unit: 2111

Regarding claim 4, Johnson et al. describe:

A receive machine (Figure 7, item 707) coupled between the host processor and
50 the bus controller (where a receive machine may be interpreted as a message data
register (MDR); column 12, lines 17 – 32). Examiner notes that the MDR functions as
both a send and receive machine, as described in column 14, lines 28 – 31).

A second FIFO buffer coupled to the receive machine and coupled between the
host processor and the bus controller (where a FIFO buffer may be interpreted as a
55 response queue; Figure 7, item 514; column 12, lines 10 – 14).

Regarding claim 12, Johnson et al. teach a computer system comprising:

A system bus implemented in accordance with an Inter-IC bus specification
(Figure 4, 7, item 310; column 7, lines 10 - 12);

60 A bus controller coupled to the system bus (where a bus controller may be
interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65);

A send machine (Figure 7, item 707; where a send machine may be interpreted
as a message data register (MDR); column 12, lines 17 – 32) coupled between a host
processor (Figure 4, item 200) and the bus controller (Figure 7, item 312), the send
65 machine comprising means for transmitting the plurality of bytes over the system bus
without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines
40 - 51 tell of how a messages is placed into the request buffer and processed by the
interface, but note that the interrupt may not be generated until a response is received
from the I2C device).

70 A first first-in first-out (FIFO) buffer (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 – 10) coupled to the send machine (Figure 7, item 707) and coupled between the host processor (Figure 4, item 200) and the bus controller (Figure 7, item 312) over a first internal bus (Figure 7, item 226) but not over the system bus (Figure 7, item 310), the first FIFO not being coupled
75 to the send machine over the first internal bus, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but
80 note that the interrupt is not generated until a response is received from the I2C device).

 A receive machine (Figure 7, item 707; where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32) coupled between the host processor (Figure 4, item 200) and the bus controller, the receive machine comprising means for receiving the plurality of bytes over the system bus
85 without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the
90 processor). Examiner notes that the MDR functions as both a send and receive machine, as described in column 14, lines 28 – 31).

A second FIFO buffer (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14) coupled to the receive machine (Figure 7, item 707), the second FIFO further coupled between the host processor and
95 the bus controller over a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second FIFO not being coupled to the receive machine over the second internal bus but not over the system bus, the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines
100 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

105

Regarding claim 43, Johnson et al. teach a computer system comprising:

A system bus (Figure 4, 7, item 310; column 7, lines 10 - 12);

A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65);

110 A send machine (Figure 7, item 707; where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32) coupled between a host processor (Figure 4, item 200) and the bus controller (Figure 7, item 312), the send machine comprising means for transmitting the plurality of bytes over the system bus

Art Unit: 2111

without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines
115 40 - 51 tell of how a messages is placed into the request buffer and processed by the
interface, but note that the interrupt may not be generated until a response is received
from the I2C device).

A first first-in first-out (FIFO) buffer (where a FIFO buffer may be interpreted as a
request queue; Figure 7, item 516; column.12, lines 7 – 10) coupled to the send
120 machine (Figure 7, item 707) and coupled between the host processor (Figure 4, item
200) and the bus controller (Figure 7, item 312) over a first internal bus (Figure 7, item
226) but not over the system bus (Figure 7, item 310), the first FIFO not being coupled
to the send machine over the first internal bus, the first FIFO comprising means for
receiving a plurality of bytes from the host processor without interrupting the host
125 processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues
are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes;
column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but
note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707; where a receive machine may be
130 interpreted as a message data register (MDR); column 12, lines 17 – 32) coupled
between the host processor (Figure 4, item 200) and the bus controller, the receive
machine comprising means for receiving the plurality of bytes over the system bus
without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 –
51 tell of how the response bytes are written into the response FIFO, and the if directed
135 to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean

Art Unit: 2111

that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor). Examiner notes that the MDR functions as both a send and receive machine, as described in column 14, lines 28 – 31).

- 140 A second FIFO buffer (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14) coupled to the receive machine (Figure 7, item 707), the second FIFO further coupled between the host processor and the bus controller over a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second
- 145 FIFO not being coupled to the receive machine over the second internal bus but not over the system bus, the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.
- 150 Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Claim Rejections - 35 USC § 103

155 The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all
obviousness rejections set forth in this Office action:

160 (a) A patent may not be obtained though the invention is not identically disclosed or described as set
forth in section 102 of this title, if the differences between the subject matter sought to be patented and
the prior art are such that the subject matter as a whole would have been obvious at the time the
invention was made to a person having ordinary skill in the art to which said subject matter pertains.
Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148
USPQ 459 (1966), that are applied for establishing a background for determining
165 obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
- 170 4. Considering objective evidence present in the application indicating
obviousness or nonobviousness.

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson
et al. (U.S. 6,122,758) in view of Yoshida (U.S. 5,928,372).

175 Johnson et al. fail to teach wherein the receive machine comprises checksum
generation means for generating a message checksum for a message while the
message is being received by the bus controller over the system bus.

Yoshida teaches a checksum generation means for generating a message
checksum for a message while the message is being received by the bus controller over
180 the system bus (where checksum generation means may be interpreted as data check
code generation circuits; column 11, lines 14 – 24).

Art Unit: 2111

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to combine the checksum generation means of Yoshida with the system of Johnson et al. in order to provide for a means of verifying the data transmitted across the system bus. This would have been obvious since error-free data is critical to the correct operation of a digital system.

* * *

Claims 6 – 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Feeney et al. (U.S. 6,072,781).

With regard to claim 6, Johnson et al. describe the computer system of claim 1, further comprising:

Means for receiving a message from the host processor (Figure 7, items 516, 707; column 12, lines 26 – 32);

Means for attempting to send the message over the system bus to a target device (column 15, lines 15 – 36 give an example of how a message is sent over the system bus to a target device (microcontroller)).

Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

Johnson et al. fail to describe retry means for attempting again to send the message over the communication bus to the target device if it is determined that the message was not received without errors by the target.

Feeney et al. teach retry means for attempting again to send the message over
205 the system bus to the target device without interrupting the host processor if it is
determined that the message was not received without errors by the target device
(column 16, lines 36 – 49 describe retrying messages that failed to send; column 16,
lines 36 – 49 describe retrying a message without involving the processor).

It would have been obvious to one of ordinary skill in this art at the time of
210 invention by applicant to incorporate the means of retrying failed messages as taught by
Feeney et al. into the computer system of Johnson et al for the purpose of ensuring the
delivery of messages on the communication bus.

With regard to claim 7, Feeney et al. teach the additional limitation wherein the
215 retry means comprises means for attempting again to send the message over the
system bus to the target device without interrupting the host processor if it is determined
that the message was not received without errors by the target device (column 16, lines
36 – 49 describe retrying a message without involving the processor).

220 With regard to claim 8, Feeney et al. teach the additional limitation wherein the
retry means comprises means for attempting again to send the message over the
system bus to the target device without obtaining the message again from the host
processor if it is determined that the message was not received without errors by the
target device (column 16, lines 36 – 49 describe storing the message in a FIFO in order
225 to allow the processor to move onto other tasks).

* * *

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson
230 et al. in view of Cao et al. (U.S. 5,230,044).

Johnson et al. fail to teach a busfree count means for storing a busfree count
associated with the computer system, a busfree timer for use by the computer system to
wait an amount of time specified by the busfree count prior to attempting to access the
system bus after the system bus becomes available for use, and a fair arbitration block
235 coupled between the host processor and the bus controller, the fair arbitration block
comprising arbitration means for modifying the busfree count according to a priority
signal to produce an arbitrated busfree count signal.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may
240 be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time
specified by the busfree count prior to attempting to access the system bus after the
system bus becomes available for use (where a busfree timer may be interpreted as a
“quiet slot” counter; column 4, lines 51 – 60);

245 A fair arbitration block comprising arbitration means for modifying the busfree
count according to a priority signal to produce an arbitrated busfree count signal (where

Art Unit: 2111

a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al. for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwidth (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

* * *

Claims 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Webb et al. (U.S.4,577,060).

With regard to claim 10, Johnson et al. fail to teach a byte timer coupled between the bus controller and the host processor.

Webb et al. teach a byte timer (where a byte timer may be interpreted as a no-response timer; column 13, lines 49 - 60).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Johnson et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the

Art Unit: 2111

system from wasting time sending messages to processors/terminals that are not
270 responsive (column 14, lines 11 – 19).

With regard to claim 11, Webb et al. teach the additional limitation wherein the
byte timer (interpreted as a no-response timer) comprises means for determining
whether the host processor has failed and means for generating a signal indicating
275 whether the host processor has failed (where a host processor may be interpreted as a
terminal; where generating a signal indicating the processor has failed may be
interpreted as marking a terminal as being “offline” or “down”; column 13, line 49 –
column 14, line 30).

280 * * *

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson
et al. in view of Feeney et al., Cao et al., and further in view of Webb et al.

Johnson et al. teach a computer system of claim 12 further comprising:

285 Means for receiving a message from the host processor (Figure 7, items 516,
707; column 12, lines 26 – 32);

Means for attempting to send the message over the system bus to a target
device (column 15, lines 15 – 36 give an example of how a message is sent over the
system bus to a target device (microcontroller)).

290 Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

Johnson et al. fail to describe a retry means, a busfree count means, a busfree count timer, a fair arbitration block, and a byte timer.

Feeney et al. teach retry means for attempting again to send the message over
295 the system bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe retrying messages that failed to send; column 16, lines 36 – 49 describe retrying a message without involving the processor).

It would have been obvious to one of ordinary skill in this art at the time of
300 invention by applicant to incorporate the means of retrying failed messages as taught by Feeney et al. into the computer system of Liu et al for the purpose of ensuring the delivery of messages on the communication bus.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may
305 be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the system bus after the system bus becomes available for use (where a busfree timer may be interpreted as a “quiet slot” counter; column 4, lines 51 – 60);

310 A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where

Art Unit: 2111

a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al., Elliot, and Feeney et al, for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwidth (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

Webb et al. teach a byte timer comprising means for determining whether the host processor has failed and means for generating a signal indicating whether the host processor has failed (where a byte timer may be interpreted as a no-response timer; column 13, lines 49 – 60; where a host processor may be interpreted as a terminal; where generating a signal indicating the processor has failed may be interpreted as marking a terminal as being “offline or “down”; column 13, line 49 – column 14, line 30).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Johnson et al., Feeney et al., and Cao et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the system from wasting time sending messages to processors/terminals that are not responsive (column 14, lines 11 – 19).

* * *

335

Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Feeney et al., Cao et al., and Webb et al.

Regarding claim 42, Johnson et al. teach a computer system comprising:

A system bus implemented in accordance with an Inter-IC bus specification

340 (Figure 4, 7, item 310; column 7, lines 10 - 12);

A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 - 36, 61 - 65);

A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a
345 message data register (MDR); column 12, lines 17 - 32), the send machine comprising means for transmitting the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 - 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

350 A first first-in first-out (FIFO) buffer (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10) coupled to the send machine (Figure 7, item 707) and coupled between the host processor (Figure 4, item 200) and the bus controller (Figure 7, item 312) over a first internal bus (Figure 7, item 226) but not over the system bus (Figure 7, item 310), the first FIFO not being coupled
355 to the send machine over the first internal bus, the first FIFO comprising means for

Art Unit: 2111

receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but
360 note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the receive machine comprising means for receiving the plurality of bytes over the system bus without interrupting the host
365 processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

370 A second FIFO buffer (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14) coupled to the receive machine (Figure 7, item 707), the second FIFO further coupled between the host processor and the bus controller over a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second
375 FIFO not being coupled to the receive machine over the second internal bus but not over the system bus, the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines

Art. Unit: 2111

18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.

380 Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Means for receiving a message from the host processor (Figure 7, items 516, 707; column 12, lines 26 – 32);

385 Means for attempting to send the message over the system bus to a target device (column 15, lines 15 – 36 give an example of how a message is sent over the system bus to a target device (microcontroller)).

Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

390 Johnson et al. fail to describe a retry means, a busfree count means, a busfree count timer, a fair arbitration block, and a byte timer.

Feeney et al. teach retry means for attempting again to send the message over the system bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device
395 (column 16, lines 36 – 49 describe retrying messages that failed to send; column 16, lines 36 – 49 describe retrying a message without involving the processor).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the means of retrying failed messages as taught by

Feeney et al. into the computer system of Johnson et al. for the purpose of ensuring the
400 delivery of messages on the communication bus.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may
be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time
405 specified by the busfree count prior to attempting to access the system bus after the
system bus becomes available for use (where a busfree timer may be interpreted as a
“quiet slot” counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree
count according to a priority signal to produce an arbitrated busfree count signal (where
410 a busfree count may be interpreted as an arbitration count number; column 5, lines 59 –
64).

It would have been obvious to one of ordinary skill in this art at the time of
invention by applicant to incorporate the busfree count and busfree timer as taught by
Cao et al. into the system of Johnson et al. for the purpose of providing arbitration
415 amongst devices on the system bus. This would have been obvious since Cao et al.
teach that their invention provides for more efficient use of bus bandwidth (column 10,
lines 25 – 59), along with permitting data communication with a very small probability of
data collisions (column 4, lines 32 – 34).

Webb et al. teach a byte timer comprising means for determining whether the
420 host processor has failed and means for generating a signal indicating whether the host

Art Unit: 2111

processor has failed (where a byte timer may be interpreted as a no-response timer; column 13, lines 49 – 60; where a host processor may be interpreted as a terminal; where generating a signal indicating the processor has failed may be interpreted as marking a terminal as being “offline or “down”; column 13, line 49 – column 14, line 30).

425 It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Johnson et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the system from wasting time sending messages to processors/terminals that are not
430 responsive (column 14, lines 11 – 19).

* * *

 Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson
435 et al. in view of Cao et al.

 Regarding claim 44, Johnson et al. teach a device for use in a computer system including a system bus (Figure 4, 7, item 310; column 7, lines 10 - 12) and a bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65), the device comprising:

440 A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the send machine comprising

Art Unit: 2111

means for transmitting the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer coupled to the send machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32), the receive machine comprising means for receiving the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

465 A second FIFO buffer (where a FIFO buffer may be interpreted as a response
queue; Figure 7, item 514; column 12, lines 10 –14) coupled to the receive machine
(Figure 7, item 707), the second FIFO further coupled between the host processor and
the bus controller over a second internal bus (interpreted as the “short” bus that
connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second
470 FIFO not being coupled to the receive machine over the second internal bus but not
over the system bus, the second FIFO buffer comprising means for receiving a plurality
of bytes from the bus controller without interrupting the host processor (column 14, lines
18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the
response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.
475 Examiner interprets this to mean that the device driver may also *not* be directed to
assert an interrupt, and therefore the reference teaches the limitation of receiving the
plurality of bytes without interrupting the processor).

Johnson et al. fail to describe a busfree count means, a busfree timer, and a fair
arbitration block.

480 Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may
be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time
specified by the busfree count prior to attempting to access the system bus after the
485 system bus becomes available for use (where a busfree timer may be interpreted as a
“quiet slot” counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al. for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwidth (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

Response to Arguments

Applicant's arguments with respect to claims 1 – 13 and 42 – 44 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not

Art Unit: 2111

510 mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

515 Any inquiry concerning this communication or earlier system from the examiner should be directed to Matthew D. Spittle whose telephone number is (571) 272-2467. The examiner can normally be reached on Monday - Friday, 8 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mark Rinehart can be reached on 571-272-3632. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

520 Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should
525 you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

530

MDS



MARK H. RINEHART
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100